1989

Precedence ambiguity in logical processing : human concept formation vs computer information processing

Sophie Schonka
San Jose State University

Order Number 1337844

Precedence ambiguity in logical processing: Human concept formation vs computer information processing

Schonka, Sophie, M.A.

San Jose State University, 1989

PRECEDENCE AMBIGUITY IN LOGICAL PROCESSING: HUMAN CONCEPT FORMATION VS COMPUTER INFORMATION PROCESSING

A Thesis

Presented to

the Faculty of the Department of Psychology

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Arts

by

Sophie Schonka

May, 1989

APPROVED FOR THE DEPARTMENT OF

PSYCHOLOGY

(Kevin Jordan, Ph.D.)

(Robert Cooper, Ph.D.)

(Stanley Taylor, Ph.D.)

APPROVED FOR THE UNIVERSITY

ACKNOWLEDGEMENTS

The author would like to acknowledge the following people for their assistance in achieving this goal: Jim Boyle, Alan Neal, and William Emmons, my management team at IBM, who have supported me through graduate school and the entire thesis process: Stanley E. Taylor, who has been my mentor through graduate school and has helped me with innumerable iterations of this thesis; R. James Evey, who read many revisions of this thesis, and provided me with support, inspiration, and many valuable suggestions for improvement; Richard Schonka, who has always been there with continuous encouragement and support.

Table of Contents

List of Tables

Precedence Ambiguity in Logical Processing:
Human Concept Formation vs. Computer Information Processing

Sophie Schonka

San Jose State University

Running Head: PRECEDENCE AMBIGUITY

Abstract

User difficulty in human-computer interaction was thought to occur because human logic is based on concept formation and conceptual information processing and computer logic is based on mathematical algorithms. Such difficulties have plagued query languages designed to allow users to extract information from databases. A study was conducted using SQL, a formal query language, which processes **and** before **or**, to learn if people process information consistent with conceptual theory or with mathematical theory. Subjects read SQL queries with **and** and **or** varied, and selected either conceptual or boolean reports as the reports expected based on their understanding of each query. Statistically significant results indicate subjects organize information conceptually and do not organize information mathematically, even though they may be aware of mathematical processes for problem solving, as some of the subjects indicated.

Precedence Ambiguity in Logical Processing:

Human Concept Formation vs. Computer Information Processing

The term "human-computer interaction" implies that a human is attempting to accomplish a goal or perform a task by communicating with a computer. For effective communication to take place, these two entities must understand each other. However, that understanding does not always occur, which results in frustration and a lack of productivity on the part of the user. The difficulties people experience have stimulated software product developers to design systems which are more usable; but to date, many problems remain unresolved and continue to plague users. A better understanding of the source of these problems is needed if adequate solutions are to be found.

One area which developers have failed to address satisfactorily is the user's perspective when working with software products. Part of the reason may be that the human thought process is so nebulous that it is difficult to model or predict. The human communication process is based on flexible skills learned in a human-to-human communication environment (Lakoff, 1987). When people interact with a computer, the knowledge they invoke is the same knowledge they use in natural communication and they expect the communication with a computer to be the same as it is with another person (Snell, 1980). For example, people have learned to understand ambiguous communication. Consider the question, "Do you want to eat Chinese food or Pizza for lunch?" A person will interpret this question as one asking for a choice to be made between the two cuisines, and will make a selection. On the other hand, the computer is a rigid machine which processes instructions in a predetermined, mathematical order (Shneiderman, 1980). Thus, the computer will not interpret the question as a choice of one item or the other. It would interpret the question as requiring an answer of "yes" if the person wanted either cuisine. The incongruence between these two methods of information processing can account for many of the discrepancies experienced when humans and computers interact.

Developers in all phases of human-computer interaction have reported user difficulty with their products. In the domain of relational databases, query languages are the tools people use to submit requests to the computer to extract database information. Past studies have found that database query languages are difficult to use (Ogden, 1986; Reisner, 1981; Thomas and Gould, 1975; Williams, 1984; Zloof, 1975). In reading and writing queries, users tend to structure them in the same way as they communicate with other people, rather than structure them based on how a computer will interpret them. This study examines the issue of precedence, pertaining to the boolean connectives **and** and **or**, as the possible cause for user difficulty when creating and interpreting Structured Query Language (SQL) queries. Written SQL queries were presented to users, and their interpretations of them were evaluated to learn if people organize information differently from computers. If users logically organize and process information differently from computers, then this difference could account for the problems people experience when working with SQL queries.

Computer background

Today the computer industry is leading an information explosion. One of the problems this introduces is designing computers which are relatively easy to use. Some years ago when computers were new and provided only complicated, electronic data processing, computer users were a well-defined audience of trained professionals. The system designers knew their users, and designed specifically for their audience. Those users expected to spend a great deal of time understanding how to use the computer and accomplish a goal. Today's computer users do not come from a homogeneous programming background. Rather, they have a wide range of backgrounds, skills, and expectations. The wide diversity of people using computers creates a problem that computer designers never faced in the past. However, today's users do not want to work for the computer but want the machine to work for them. It is the role of Human Factors engineers to reconcile this inharmony by tailoring computer interfaces to be more compatible to human thinking.

Computers are machines that can perform complex functions based on mathematical logic. They process information and return precise answers to complex questions faster than any human could ever hope to achieve. The logic computers use, however, is not the same as that which humans employ. Computers use boolean algebra or first-order predicate calculus as logic to manipulate information (Shneiderman, 1980). They generally process information in a non-adaptive, algorithmic manner that does not deviate based on context. Each statement issued to a computer is interpreted in an exact, predetermined order.

Databases and Query Languages

One of the computer's greatest attributes is its ability to store, manipulate, and update tremendous amounts of information, which is done through databases and database management systems. Database information extends from the supermarket check-out line to bank transactions and airline reservations. The amount of data now dedicated to databases is measured in billions of bytes in typical large information centers.

Database information is accessed by query languages. A variety of models for database query languages has been proposed, but three have emerged as the most popular: the hierarchical, network, and relational models (Reisner, 1988). Of the three, the relational model is the fastest growing in industry today. The relational model is based on the ideas of E. F. Codd (1970). It requires data and relationships to be shown as tabular structures. Data are organized in columns and rows within each table. Every column has a name which identifies the information residing in that column. Each row contains unique pieces of information, known as tuples. A column of data in one table can be matched to a column of data in another table, thereby allowing a "relationship" between tables. Users can then join tables to obtain extended information. The joining ability eliminates the need for storing duplicate information.

The database used in this study contains information for restaurants in the San Francisco bay area. It illustrates the structure of a relational database. Table 1 shows a sample of the database information.

SQL

A user must write a short program known as a query to retrieve information stored from a relational database. One of the most common query languages available today is a boolean based query language known as SQL (Structured Query Language). Boolean query languages permit the use of certain logical operators including **and** and **or**. SQL is currently the industry's standard query language for relational databases and has been studied extensively over the years (Boyle, Evey & Neal, 1980; Greenblatt & Waxman, 1978; Ogden, 1986; Reisner, 1977; Shneiderman, 1980; Welty & Stemple, 1981).

Query languages vary in their ability to provide comprehensive coverage for all potential queries. SQL is the most developed and tested language in the query field. In spite of its in-depth evaluations, however, problems continue to occur and plague users. Many queries that can be written with SQL are extremely difficult to compose and comprehend. Some common, useful, and simple queries present enormous problems to the users. These problems occur, in part, because people have difficulty understanding boolean logic structures.

Name[1]	City	Cuisine	Price
Arthur's Grill	Santa Clara	Seafood	Very Expensive
Asia Garden	San Jose	Chinese	Inexpensive
Bella's	San Francisco	German	Moderate
Maxi's	Mountain View	Continental	Moderate
The Rogue	Los Gatos	French	Expensive

Table 1. Sample of Database

[1] The names of these restaurants were chosen from local area restaurants to convey a realistic database, but they are not meant to represent real information.

A report by Ogden, Korenshtein, and Smelcer (1986) summarized some of the cognitive problems users experience when writing SQL queries. For example, users omitted the function "GROUP BY." In language, the concept of aggregation implies grouping so we don't have to say it; but when we communicate with a computer, with SQL, we have to define this structure. For example, if we ask for a department number and average salaries, we understand that we want the average salary by department. In SQL, we must say this, i.e., Select Dept, Avg Salary, ... Group by Dept. Users forgot to put quotes around variables. In language, the context identifies the data type. SQL requires quotes to distinguish character strings from numbers. Users had difficulty distinguishing between the functions "where" and "having." In verbal communication, people can use these two words as synonyms and be understood by the listener. But in SQL, the first restricts selection of individual rows, and the second, data from the rows which have been grouped.

Precedence

Users generally have many problems understanding how to use query languages when they attempt to work with database information. One interesting problem has to do with precedence. For example, consider the following columns of information available in a table of San Francisco Bay Area restaurants:

NAME CITY CUISINE PRICE

If users want to eat pizza in San Jose or Morgan Hill, they would write a query to obtain that information from the database. Using SQL, the query could look like this:

```
Select  Name, City, Cuisine
From Restaurant_Database
Where City = 'San Jose' or City = 'Morgan Hill'
and Cuisine = 'Pizza'
```

This query seems logical to most people. A person interprets this query as a request for a list of restaurants in either San Jose or Morgan Hill, which serve pizza. However, the information that this query will return to the user will surprise the person. This query will not be parsed by categorical logic, where "pizza" is intended to apply to both San Jose and Morgan Hill, but will be parsed by the computer's precedence-based logic which ignores content. Precedence logic dictates that **and** is processed before **or**. Therefore, the computer will understand this query as follows:

```
Select Name, City, Cuisine from Restaurant_Database
Where (City is San Jose) or (City is Morgan Hill and Cuisine is Pizza)
```

In this example, the computer's logic considers **or** as an inclusive union and **and** as an exclusive intersection. Its interpretation of this query is to present the results of both parenthetical situations as the correct response. Therefore, this query will give the user a report that shows all the restaurants in San Jose, and those in Morgan Hill which serve pizza.

A user typically will not understand what happened, and will try again to get the report they want. This problem is not experienced by naive computer users exclusively. Knowledgeable users often forget to think in boolean logic and wonder what happened when they receive a report they did not expect. An SQL expert who was recruited to participate in this study demonstrated this difficulty.

Ogden and Kaplan (1987) studied the use of **and** and **or** in a natural language computer interface. They found that **or** was almost always used correctly to indicate a union, but **and** was used to indicate both union and intersection. These results led them to suggest that natural language interfaces should be built to recognize ambiguous input and should prompt users for correct clarification of their intentions.

The problem of precedence is one which appears to occur because people expect computer-human communication to be like human-human communication. Therefore, the problem of generating correct queries may be attributed to the fact that people and computers operate using different logical methods.

Computers organize information based on particular mathematical algorithms. SQL uses boolean logic (Shneiderman, 1980). Boolean logic dictates the computer to manipulate boolean operators, such as **and** and **or**, and parse a query based on the precedence of these two operators. This means **and** gets processed before **or**. The results of this parsing do not consider the conceptual information surrounding each of these values. This difference in precedence processing can account for some of the difficulty users experience when working with query languages.

Categorical Logic

The word "category" implies that items are organized together based on what they have in common (Rosch, 1975). We use categories to logically organize our physical and spatial world, and to understand relationships among objects and concepts. We employ categories whenever we reason about types of things. Whenever we utter or listen to a string of words, we are employing many categories: categories of speech sounds, words, phrases, and conceptual categories. We would be unable to function in either the physical or social world without an ability to categorize.

People do not process information linearly, precisely, or consistently. People tolerate ambiguity. They use cues from the environment and they can understand homonyms, gestures and inflections. The communications skills people employ are learned in an environment where people communicate with other people (Rosch, 1975). Children learn from their surroundings. Communication is taught to them by the people with whom they come in contact. The ease with which people use language gives one the illusion that language communication is basically a simple process, not involving much knowledge (Lakoff, 1987).

The errors made in the query shown earlier regarding restaurants in Morgan Hill suggests that the users may have ignored the boolean precedence because they believed that conceptually related material would be processed first, that is, given precedence. Thus, their logic thought that the union of Chinese cuisine and pizza took precedence.

For the formally trained, SQL itself has recognized this problem in part. Thus, the more typical means for indicating a union within a category would be by using an "IN" statement such as "Where Cuisine in (Chinese, Pizza)." But this cannot always be done in SQL, particularly where conceptually similar material is distributed over several columns.

Current Research

Although SQL has undergone a great deal of experimentation and evaluation, the role of categorical grouping in the interpretation of precedence has not been studied. I hypothesized that people bring with them conceptual knowledge of the natural world when they interact with the artificial world of computers. Users do not switch to boolean logic as required by computers to accomplish their task. It was predicted that users would expect results from the set of queries that were consistent with conceptual logic, and not boolean logic.

Sixteen tasks written in SQL were presented to users. The position of **and** and **or** was manipulated in each of the tasks. In each task, users chose from three types of reports that they thought the query could produce; one based on boolean logic, one based on conceptual theory, and one filler. The filler report was used to determine if the subjects employed a strategy in report selection, or if they randomly selected any of the three reports. Users were asked to select which report(s) matched their understanding of each query. If users interpreted queries conceptually, then they would be expected to select the conceptual reports. If they interpreted the queries using boolean logic, then they would be expected to select the boolean reports. Two experiments, differing only in how the reports were presented, were conducted to determine subjects' strategy in interpreting queries.

Experiment 1

Method

Subjects. Subjects were recruited from employees at IBM Corporation. A total of 33 subjects participated in the study; 8 pilot subjects, 13 in Experiment 1 and 12 in Experiment 2. Subjects were randomly assigned to one of four orders of task presentation. The subjects were screened to meet the criteria of having no SQL knowledge. This was done specifically to eliminate people already familiar with boolean logic. An additional subject was an SQL expert. This subject is excluded from the grouped results and is discussed independently in Experiment 1.

Tasks. Sixteen SQL queries were written to obtain information from a restaurant database built for this study. The queries were written so that each statement of the query and each **and** and **or** began on a new line, thereby eliminating any visual grouping of information. The tasks were first given to eight pilot subjects who completed them on paper. After the pilot subjects completed the study, the queries were rewritten to address one problem encountered by these subjects. The pilot subjects' felt the word SELECT was misleading. They thought they were to select individual items from each report, rather than select an entire report. Therefore, the word SHOW was substituted for the word SELECT. For example, the correct SQL statement to display rows from a database is:

Select name, city, cuisine

The SQL queries in this study were rewritten as follows:

Show name, city, cuisine

Following is a sample of the task queries:

Show Name, City, Cuisine, Price

Where Cuisine is Italian

Or Cuisine is French

And City is San Francisco

Or City is San Jose

Material. Each task contained one query and a packet of three reports labeled "A," "B," or "C." The alphabetic characters were randomly assigned to each of the three reports, and the reports were randomly ordered in the packet. The reports were hard-copy which enabled subjects to examine the reports while reading the task which appeared on their terminal screen. One report was the complete boolean report as produced by SQL. The second report was a categorical report generated by placing parentheses around the query to be consistent with categorical grouping. The third report was a filler report which was neither boolean nor conceptual in nature. The boolean report included the conceptual report, so it was the longer report in every task.

To account for the conceptual reports always being shorter than the boolean reports, the filler reports were designed to be the same length as the boolean reports in half the tasks, and the same length as the conceptual reports in the other half of the tasks. Since the boolean reports naturally contained conceptual data, the data in each type of report were not exclusive. Additionally, the filler reports contained some data from both the boolean and conceptual category, and also contained information that satisfied neither of these categories.

Apparatus. The tasks were installed on an IBM VM system. Subjects used a IBM 3279 color terminal. A REXX[1] program presented the tasks to the subjects in the designated order and also accepted the subjects' response for each task. The time on task was recorded by VM through the REXX program. The subject room was equipped with overhead and side mounted cameras and microphones. This equipment was pointed out to the subjects and its use was fully explained to each of them. Subjects were observed during the study so the observer could assist them in case of system failure, or if subjects experienced unusual difficulty.

---

[1] Restructured Extended Executor language similar to PL/I which runs on VM CMS

Experimental Design. A database was built that contained 200 entries with four columns of data representing restaurant information. The words **and** and **or** were the variables systematically manipulated in the queries. The database columns were identified with alphabetic characters to facilitate mapping them to the experimental design. Table 2 shows the database columns (with their identifying characters) and variables in four conditions.

Each subject completed 16 tasks. There were four conditions, and four tasks within each condition. Each of the four conditions was identified with a Latin character, and each of the four tasks within each condition was identified with a Greek character. A Greco-Latin square design was used to order and counterbalance the tasks in the blocks. A Latin square design was used to order the presentation of blocks, resulting in four repetitions of task order. Table 3 shows the experimental design with the tasks and conditions in the counterbalanced order. The blocks were then counterbalanced in order of presentation to the subjects.

Procedure. Subjects were tested individually. They arrived at the IBM Human Factors Laboratory where they were given a brief tour, and also a general description of Human Factors work. A pre-test questionnaire was administered to learn each subject's computer experience, and to confirm that each subject did not know SQL. The concept of databases was explained to them and a copy of the restaurant database was shown to them. Each column of information was pointed out and explained to them. Subjects were told that

Con.	Col. A	Var.	Col. B	Var.	Col. C	Var.	Col. D
1	City	And	Cuisine	Or	Cuisine	And	Price
2	City	Or	City	Or	City	And	Cuisine
3	Cuisine	Or	Cuisine	And	City	Or	City
4	Cuisine	Or	Cuisine	And	Price	And	City

Table 2. Experimental Conditions

Block 1	Bδ	Dβ	Aγ	Cα
Block 2	Aα	Cγ	Bβ	Dδ
Block 3	Dγ	Bα	Cδ	Aβ
Block 4	Cβ	Aδ	Dα	Bγ

Table 3. Greco-Latin Square Experimental Design

query languages are designed to extract specific information from databases. The mechanics of how SQL works were reviewed with them. Subjects were not told about SQL syntax or how it specifically worked. At the conclusion of each subject's orientation, they were asked to sign an informed consent form.

The instructions to the subjects were on the terminal screen at the beginning of the session. A subject's task was to press ENTER to see the first query, at which point the clock started. The subject was instructed to read each query. The subject then examined a packet of reports, and decided which one of the three best matched their understanding of the query. When the subjects reached a decision, they pressed ENTER again. This stopped the clock and presented a screen on which the subjects could type their answer. After entering their answers, the subjects pressed ENTER which took them to an intermediate screen. From this panel, they pressed ENTER when they were ready to begin the next task. This sequence occurred over sixteen trials. All subjects finished in less than one hour.

Results and Discussion

The results indicate that, on the average, 91% of the time subjects expected the computer to organize information conceptually. In 9% of the cases, on the average, subjects chose the boolean reports as meeting their interpretation of the query. The percentage of reports chosen and the average time for each task is shown in Table 4. A Chi square analysis found a significant difference in the

Task	Percent Conceptual	Percent Boolean	Percent Filler	Average Time
1-1	75	25		65.3
1-2	75	25		66.3
1-3	84	16		57.5
1-4	84	16		52.4
2-1	100			35.3
2-2	92	8		43.3
2-3	92	8		58.3
2-4	100			108.3
3-1	100			34.3
3-2	92		8	78.1
3-3	92		8	103.2
3-4	75	17	8	203.8
4-1	100			38.9
4-2	92		8	45.4
4-3	100			40.3
4-4	100			40.4
Summary	91	9		89.3

Table 4. Percent of Reports Chosen Per Task

subjects' selection of conceptual reports over boolean or filler reports χ^2 (1, N = 12) = 172.26, $p < 0.005$.

These results indicate that subjects do not expect computers to organize information using boolean logic. They appear to expect computers to organize information conceptually. The queries used in this study contained information that is familiar to all people; names of restaurants, cities, cuisines, and price ranges. These are distinct categories which people do not confuse. The results show that people understood the categories, and applied their everyday knowledge to interpret the queries. There are two queries in which all subjects selected the conceptual reports which demonstrate the conceptual nature of subjects' interpretations. Query 2-1 stated:

> Show Name, City, Cuisine
>
> From Restaurant Database
>
> Where Cuisine is Mexican
>
> or Cuisine is Italian
>
> or Cuisine is Chinese
>
> and City is San Jose

The second query, 4-3, stated:

> Show Name, City, Cuisine, Price
>
> From Restaurant Database
>
> Where Cuisine is Italian
>
> or Cuisine is Seafood

and City is San Francisco

and Price is Moderate

In both examples, subjects did not consider the cuisines without assimilating all the information into their decisions. They understood that a restaurant served only one kind of cuisine, and expected any to be correct as long as the results matched all the criteria.

Query 3-4 shows an average time on task which is somewhat higher than the other tasks. This query stated:

Show Name, City, Cuisine, Price

From Restaurant Database

Where Cuisine is French

or Price is Moderate

and City is Palo Alto

or Cuisine is Seafood

This query was the only one which separated two identical categories (Cuisine is French or Cuisine is Seafood) with intervening information. There were four possible ways to parenthesize the conceptual categories for this query. The way that was selected and presented to the subjects as the conceptual report did not match the standard conceptual report of the other queries. The conceptual report listed French or Moderate restaurants in Palo Alto, and all the Seafood restaurants in the database. This may have confused subjects since they had become accustomed to simpler conceptual reports. A simple conceptual report would have reorganized the data of the query to have shown only the French, Moderate, or Seafood restaurants in Palo Alto. Subjects may have spent the extra time on this query trying to find one report which fit this conceptual model. In spite of the fact that the conceptual report they were given may not have been what they expected, 75% of the subjects still selected it as their correct response. It appears that the inclination to organize the information conceptually is so strong that subjects forced the information as they understood it to fit into the report provided to them. Additionally, these subjects did not consider the boolean report correct in spite of their effort to find the right report.

After the first 12 subjects completed the study, an SQL expert was asked to participate as a subject. The procedure for this subject, subject 33, was identical to the other subjects. Subject 33 was expected to select the boolean reports associated with each query.

Subject 33 selected 81% of the conceptual reports as the reports he expected based on his understanding of each query. This information is significant in several respects. First, this subject is an SQL expert who uses the product daily. He is intimately familiar with boolean logic, and yet his conceptual intuitiveness overrode his mathematical knowledge in deciding which report to select. Second, when he observed the first task, and noticed the absence of parentheses which are used to define grouping of values, he mentioned their absence and asked if he was to interpret each query as he knew the system would parse it.

Therefore, he was aware of the problem immediately. He commented that he was organizing the task information using boolean logic. Therefore, he believed that the conceptual reports he was selecting were actually the boolean reports. Thirdly, he had the boolean reports, which he reviewed for each task, but failed to recognize them as the correct SQL reports generated by each query.

The information collected from Subject 33 is quite strong in favor of conceptual organization of information. The ability to suppress conceptual criteria when organizing information seems very difficult to do. It appears that even if one is aware of boolean logic, it is difficult to maintain a mathematical approach to problem solving.

The results from the first 12 subjects presented sufficient results to not require further testing. However, at this time, it seemed two issues existed which were unresolved. The first issue was the length of the reports. In each task, the conceptual report was shorter than the boolean report. Since 91% of the conceptual reports were selected, one could argue that subjects were selecting the shorter reports. The second issue was the conceptual report for task 3-4. This report was not structured as simply (conceptually) as the remainder of the conceptual reports. This could account for the additional time spent on this task. Therefore, it was decided to restructure the conceptual report for this task to make it similar to the remainder of the reports. A second experiment was conducted to control for these two issues.

Experiment 2

Method

Subjects. Twelve subjects participated in Experiment 2. With the exception of the following categories, Experiment 2 was conducted the same as Experiment 1.

Material. In Experiment 2, the boolean and conceptual reports were derived by running the SQL query for each task. Once this report was generated, the data was separated into two categories, one that fit a boolean solution but not a conceptual solution, and one which fit a conceptual solution exclusively. Four reports were then created for each task from this data, with each report containing ten lines of information. Additionally, two filler reports were designed for each task. These reports were also ten lines long. The data in these reports were exclusive of the other two categories, that is, they each contained information not found on the other reports. Tables 5, 6, 7, and 8 show a task and the three reports associated with that task as used in the study for Experiment 2.

These reports represent the structure of the reports given to the subjects. The conceptual report organized the information conceptually. That is, it selected French and Italian restaurants within San Jose. The boolean report gave subjects boolean data; any Italian restaurant in the database. The filler report contained data which did not match either boolean or conceptual organization. It gave the subjects extraneous information.

Show Name, City, Cuisine

Where City is San Jose

And Cuisine is French

or Cuisine is Italian

Table 5. Query Number 1-1

Name	City	Cuisine
Agnew's Grill	San Jose	French
Amadeus	San Jose	French
Antonio's	San Jose	Italian
Arc En Ciel	San Jose	French
Cadillac Bar	San Jose	Italian
Cafe Cipriani	San Jose	Italian
Cafe D'Or	San Jose	Italian
Cafe Riggio	San Jose	Italian
Chiantis	San Jose	Italian
Ciao	San Jose	Italian

Table 6. Conceptual Report for Query 1-1

Name	City	Cuisine
Adriana's	San Francisco	Italian
Avanti Pasta	Los Altos	Italian
Baltic	Berkeley	Italian
Bella Mia	Saratoga	Italian
Cafe Luciano	San Francisco	Italian
Da Sandro	Berkeley	Italian
E'Angelo	Sunnyvale	Italian
Firenzia	Campbell	Italian
Giramonti	San Francisco	Italian

Table 7. Boolean Report for Query 1-1

Name	City	Cuisine
Valley Inn	San Geronimo	California
The Abby Restaurant	St. Helena	Continental
The Caprice	Tiburon	Continental
Depot Hotel 1870	Sonoma	Continental
Divenuta	Napa	Continental
Inn at the Tides	Bodega Bay	Continental
Lark Creek Inn	Larkspur	Continental
Petit Maure	Novato	Continental
Manka's Czech Restaurant	Inverness	Czech
Pelican Inn	Muir Beach	English

Table 8. Filler Report for Query 1-1

Experimental Design. This design afforded us two additional controls. First, all reports were exactly the same size. Secondly, because the conceptual reports are a subset of the boolean reports, four reports always matched the boolean criteria. However, four reports did not always match the conceptual criteria. Therefore, if subjects overlooked the boolean reports, the results would more strongly support the theory that subjects were organizing information conceptually. Reports were randomly labeled from "A" to "F." Subjects were allowed to select any number of the reports, from zero to six, as their choice of correct reports. The number of boolean logic reports and conceptual reports was distributed in a counterbalanced order among tasks. This was done so subjects would not find a pattern in report selection. Table 9 shows the distribution of the reports.

Results and Discussion

Subjects were told they could pick any number of the six reports they felt satisfied their selection criteria. Table 10 shows the percentage of each type of reports that elicited a response from the subjects, based on the total number of selections made per task. Also shown in Table 10 is the average time for each task.

A Chi square analysis found a statistically significant difference between the percentage of boolean reports selected and the percentage of conceptual reports selected χ^2 (1, N=12) = 55.7, $p < 0.0001$, with 84% of the reports selected being conceptual, 15%, boolean, and 4% filler.

No. of Tasks	Conceptual Reports	Boolean Reports	Filler Reports
1	0	4	2
4	1	3	2
6	2	2	2
4	3	1	2
1	4	0	2

Table 9. Distribution of Reports

Task	Percent Conceptual	Percent Boolean	Percent Filler	% None Chosen	Average Time
1-1	79	21			73.4
1-2	75	25			75.1
1-3	71	29			86.8
1-4	40	57	3		96.7
2-1	100				52.9
2-2	88	8	4		58.2
2-3	95	5			77.4
2-4	90	3		7	131.4
3-1	80	20			45.8
3-2	100*				76.2
3-3	97	3			77.5
3-4	79	21			107.7
4-1	82	18			81.9
4-2	84	8		8	81.2
4-3	100				68.9
4-4		15		85**	77.8
Summary	84	15	4		79.4

Table 10. Experiment 2 - Percent of Reports Chosen Per Task

* This task contained no boolean reports
** This task contained no conceptual reports

I believe the filler report chosen for task 1-4 was selected accidentally. The subject was looking at report "D" which was a conceptual report, and yet typed a "C" which indicated a filler report. The error probably occurred because the "D" and "C" are next to each other on the keyboard and the subject was not looking at the keyboard at the time of the entry.

In all but three tasks (3-2, 3-3 and 3-4), the time-on-task was longer for Experiment 2 than Experiment 1. One reason could be that the subjects in Experiment 1 had fewer reports to consider in making their selection, and they were told to select only one report. The subjects were observed while working on the tasks. In both experiments, subjects seemed to examine the reports for exceptions to their criteria rather than make sure everything met their expectations. If a subject encountered a single item that did not fit their expectations, they disregarded the report, and looked at the next one. Therefore, in Experiment 1 they could quickly reject two reports and search only one thoroughly, whereas in Experiment 2 they examined the reports more carefully before excluding them.

In task 3-2, the time on task for Experiment 2 is less than three seconds shorter than Experiment 1. It is difficult to account for the shorter time spent on task 3-3. There was nothing immediately obvious about this task that could have affected the time-on-task differently from the other tasks. In task 3-4, the conceptual reports were rearranged to reflect a simpler conceptual nature than they did in Experiment 1. There were two conceptually organized reports for this task, and both showed restaurants only in Palo Alto. I think this simplification assisted the subjects in selecting their choices more quickly in Experiment 2.

An additional analysis was conducted to analyze the number of reports chosen within each distinct report category. Table 11 shows what percentage of each type of reports available were selected per task.

A 2x2 contingency table was constructed to determine if a statistically significant difference existed between the proportion of boolean and conceptual reports selected. Table 12 shows the contingency table used to conduct this analysis.

A χ^2 analyses comparing these proportions found statistical significance between the two; χ^2 (1, N=12) = 296.04, $p < 0.0001$. Of the conceptual reports, 94% were selected; of the boolean reports, 16% were selected. This analysis indicates that subjects purposely chose the conceptual reports and purposely avoided selecting the boolean reports.

General Discussion

The results from both experiments indicate that subjects organize information conceptually. They do not appear to organize information mathematically even though they may be aware of mathematical processes for problem solving. These findings are consistent with Ogden and Kaplan's (1987) study which found that subjects had difficulty using **and** and **or** correctly. They also support a report by de Stricker (1988) who discusses user problems with

Task	Percent Conceptual	Percent Boolean
1-1	96	25
1-2	100	29
1-3	96	37
1-4	100	47
2-1	97	0
2-2	92	8
2-3	94	17
2-4	72	8
3-1	100	8
3-2	94	*
3-3	89	8
3-4	92	25
4-1	96	21
4-2	92	3
4-3	100	0
4-4	**	4
Summary	94	16

Table 11. Experiment 2 - Percent of Reports Selected Within Each Category

* This task contained no boolean reports

** This task contained no conceptual reports

	Boolean	Conceptual	
Choose	124	355	479
Not Chosen	260	29	289
	384	384	768

Table 12. Contingency Table

boolean logic in menu driven interfaces. But these results extend our understanding of that difficulty. It appears that conceptual information affects how precedence is understood.

Several subjects in both experiments commented on the structure of the queries and the use of logic precedence. One subject told me: "In math, you put parentheses around the items you want to group. You don't put parentheses around 'Mexican' and 'Inexpensive'; they are two different things." Another subject asked me how the **and** and **or** worked, and if SQL used parentheses. He was told to organize the information as he chose. His response was that he would operate strictly on precedence and process **and** before **or**. He said he was "mind-set" this way because he had been programming for the last year. This subject selected 93% of the conceptual reports presented to him and only 58% of the boolean reports. It appears that he was unable to follow his own intentions, and that the tendency for conceptual grouping was so strong that it overrode his intention to give precedence to **and.** This subject supports the results obtained from Subject 33 who demonstrated the same difficulty in selecting the boolean reports even though he is an SQL programmer and intimately familiar with boolean processing.

Within cognitive psychology, categorization has become a major field of study. Whenever we reason about kinds of things, we are employing categories. Most categorization is done unconsciously, and if we are conscious of it at all, it's in problematic instances (Rosch, 1975). Categorization is also done abstractly. We categorize events, spatial relationships, emotions and social relationships. Our abstract concepts include governments, illnesses, and scientific theories (Lakoff, 1987). It appears then, that categorization is crucial to every view of reason.

Several theories about categories have evolved over the years. The subjects in this study behaved according to the classical theory of category formation. This theory is the oldest category theory and stems from Aristotle's philosophy which defined a category as an abstract container containing similar items. Subjects clearly grouped items based on what they had in common. They understood that the name of a city meant they were interested in that city only, and any other parameters listed were assumed to be associated with that city. They also understood that items can be mutually exclusive. That is, they didn't expect a restaurant to be in two cities at the same time.

Although the queries were written in such a way that they had multiple interpretations, the subjects did not readily see the alternate interpretations. In performing a mental task, such as math or playing tic-tac-toe, we are conscious of the mental activity that takes place in order to accomplish the task. However, when we talk, or listen to someone talking, we do not think of the mental activity engaged in understanding speech, unless something goes wrong (Winograd, 1983). In the case of this study, the subjects never became conscious of their thought process and consequently never realized that anything went wrong. Therefore, they never examined the tasks carefully enough to see alternative interpretations.

The results of this study argue strongly for software interfaces which process information in a manner more consistent with human thinking. One solution which could be implemented to address this problem would be to design an interface with sufficient intelligence to prompt users if a query could be interpreted in more than one way. For example, if more than one boolean operator existed in a query, the system could parse the query in all possible ways and then prompt the users to select which one was their intended query.

The results present a new problem for consideration and evaluation. Careful examination of the queries found that it is possible the subjects were processing information by giving **or** precedence, rather than organizing information conceptually as argued by this thesis. A follow-on test should be designed to account for this variable. The author believes that the comments from subjects who stated an intention to give **and** precedence, or who made statements to the effect that conceptually similar things should be processed together, constitutes evidence that it is, in fact, conceptual similarity that brought about the problems.

References

Boyle, J. M., Evey, R. J., and Neal, A. S. (1980). Computer controlled comparison of two database query languages. (Tech. Rep. HFC-33). San Jose: General Products Division, Human Factors Center.

Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13, 377-387.

deStricker, U. (1988). A menu interface to formulate boolean logic - can it be done? Information Services & Use, 8, 39-46.

Greenblatt, D., and Waxman, J. (1978). A study of three database query languages. In B. Shneiderman (Ed.), Databases: Improving usability and responsiveness. New York: Academic Press.

Lakoff, G. (1987). Women, fire and dangerous things. Chicago: The University of Chicago Press

Ogden, W. (1986). Implications of a cognitive model of database query: comparison of a natural language, formal language, and direct manipulation interface. ACM SIGCHI Bulletin, October, 51-54.

Ogden, W. and Kaplan, C. (1987). The use of and and or in a natural language computer interface. Proceedings of the Human Factors Society - 30th Annual Meeting (pp. 829-833). Santa Monica, California: The Human Factors Society.

Ogden, W. C.; Korenshtein, R. and Smelcer, J. B. (1986). An intelligent front-end for SQL: Correcting user errors with Natural Language Menus

(Tech. Rep. HFC-55). San Jose: General Products Division, Human Factors Center.

Reisner, P. (1977). Use of psychological experimentation as an aid to development of a query language. IEEE Transactions on Software Engineering, SE-3, 218-229.

Reisner, P. (1981). Human factors studies of database query languages: a survey and assessment. Computing Surveys, 13, 13-31.

Reisner, P. (1988). Human factors studies of database query languages, updated (Tech. Rep. RJ6160). San Jose: IBM Almaden Research Center, Computer Science Department.

Rosch, E. (1975). Cognitive reference points. Cognitive Psychology, 7, 532-47.

Shneiderman, B. (1980). Software psychology: Human factors in computer and information systems. Boston: Little, Brown and Co.

Snell, J. L. (1987). From human cognition models to human-factored computing. IEEE Transactions on Software Engineering, SE-3, 450-454.

Thomas, J. C., and Gould, J. D. (1975). A psychological study of query by example. In Proceedings of the National Computer Conference (pp. 439-445). Arlington: AFIPS Press.

Welty, C., and Stemple, D.W. (1981). Human factors comparisons of a procedural and a nonprocedural query language. ACM Transactions on Database Systems, 6, 626-649.

Williams, M. D. (1984). What makes RABBIT run? Int. J. Man-Machine Studies, 21, 333-352.

Winograd, T. (1983). Language as a cognitive process. Reading Mass: Addison-Wesley

Zloof, M. M. (1975). Query by example. In Proceedings of the National Computer Conference (pp. 431-437). Arlington: AFIPS Press.